PEMBELAJARAN STRUKTUR ALJABAR DENGAN MENGGUNAKAN *SOFTWARE* GAP

Ema Carnia, Isah Aisah, Sisilia Sylviani

Jurusan Matematika
Universitas Padjadjaran

ABSTRAK

Strukur Aljabar sebagai salah satu mata kuliah wajib yang diberikan di Program Studi Matematika di Indonesia dirasakan sulit oleh sebagian besar mahasiswa. Sebagai salah satu alternatif untuk mengatasi hal itu maka diperkenalkan penggunaan *software* GAP (*Group, Algorithm, and Programming*) dalam proses belajar mengajar pada mata kuliah Strukur Aljabar. Selain mendukung dalam proses pembelajaran, penggunaan *software* GAP ini mendukung pemberlakuan Kurikulum Berbasis Kompetensi (KBK) di Perguruan Tinggi, khususnya di Program Studi Matematika, yang secara tidak langsung menuntut pembelajaran *Student Centered Learning* (SCL). Pengajaran dengan menggunakan *software* ini akan diujicobakan di Departemen Matematika FMIPA UNPAD. Dengan demikian diharapkan dapat memberikan motivasi untuk belajar Struktur Aljabar dengan cara yang tidak membosankan yang pada akhirnya dapat meningkatkan pemahaman mahasiswa terhadap mata kuliah tersebut. Penelitian ini baru sebatas kajian teori dan belum diimplementasikan

Kata kunci: GAP, Struktur Aljabar.

ABSTRACT

As one of the compulsory subjects given in the Mathematical Studies Program in Indonesia, Algebraic Structure is perceived as difficult by most students. One alternative to overcome it is by introducing the use of GAP software (Group, Algorithm, and Programming) in teaching and learning for Algebraic Structure course. In addition to support the learning process, the use of GAP software support the implementation of the Competency Based Curriculum in universities, especially in Mathematics Program, which indirectly requires Student Centered Learning (SCL). Teaching with this software will be tested in Department of Mathematics Faculty of Mathematical and Natural Sciences. Hence the use of it is expected to provide the motivation to learn algebraic structure in a way that is not boring, and will subsequently increase students' understanding of the course. This research is a theoretical study and has not been implemented yet.

Keywords: *Algebraic Structure, GAP.*

PENDAHULUAN

Struktur aljabar merupakan salah satu mata kuliah yang diberikan di Program Studi Matematika. Untuk jenjang S1, dalam mata kuliah tersebut dipelajari Teori Grup, *Ring*, Daerah Integral, dan Lapangan. Pada umumnya referensi Struktur Aljabar memulai dengan memperkenalkan definisi-definisi aksiomatik dan sifat-sifat dari struktur tersebut yang tidak mudah untuk dipahami oleh mahasiswa karena terlalu abstrak, sehingga sulit dibayangkan secara riil.

Berbagai upaya telah dilakukan untuk membantu mahasiswa dalam memahami serta meningkatkan kualitas pembelajaran mata kuliah tersebut. Upaya tersebut misalnya dengan diadakannya tutorial, menjelaskan materi secara detil disertai dengan contoh-contoh yang lebih riil.

Pemberlakuan Kurikulum Berbasis Kompetensi (KBK) di Perguruan Tinggi, khususnya di Program Studi Matematika, yang secara tidak langsung menuntut pembelajaran *Student Centered Learning* (SCL) menjadi tantangan tersendiri bagi dosen (Direktorat Akademik Ditjen Dikti, 2008). Hal tersebut dikarenakan pada umumnya kegiatan pembelajaran mata kuliah struktur aljabar selama ini cenderung berpusat pada dosen dan mahasiswa cenderung kurang aktif. Salah satu upaya yang dapat dilakukan untuk menjawab tantangan tersebut adalah dengan melakukan berbagai inovasi dalam metode pengajaran Struktur Aljabar. Metode pembelajaran alternatif tersebut adalah metode yang selain dapat memudahkan mahasiswa un-

tuk memahami materi-materi yang diberikan, juga dapat meningkatkan kreativitas serta meningkatkan hasil belajar mahasiswa, selain juga harus dapat memenuhi tuntutan KBK tersebut.

METODE

Penelitian ini merupakan kajian teori dan belum diimplementasikan. Penelitian ini bertujuan untuk meneliti manfaat GAP berdasarkan kajian teoritis dan literatur.

HASIL DAN PEMBAHASAN

Kajian teoritis yang kami lakukan menghasilkan beberapa informasi sebagai berikut. GAP merupakan singkatan dari *Group, Algorithm, and Programming*. *Software* GAP bersifat gratis, terbuka, dan *extensible* yang digunakan untuk komputasi pada Struktur Aljabar, khususnya yang berorde hingga. GAP pertama kali dikembangkan pada tahun 1986 di *Lehrstuhl D für Mathematik*, RWTH Aachen Jerman oleh Joachim Neubüser, Johannes Meier, Alice Niemayer, Werner Nickel, dan Martin Schönert. Versi pertama GAP yang diperkenalkan pada publik adalah versi 2.4. Sekitar 22 tahun kemudian, tepatnya pada tahun 2008, GAP mendapatkan penghargaan dari *The ACM/SIGSAM Richard Dimick Jenks Memorial Prize* sebagai *software* teknik yang unggul untuk komputasi aljabar. Hingga kini GAP masih terus dikembangkan di *University of St. Andrews*, di St. Andrews, Skotlandia, dan saat ini versi GAP yang terbaru adalah 4.6.3 yang dirilis pada 18 Maret 2013 (*The GAP Group*, 2006).

Bahasa pemrograman yang digunakan oleh GAP adalah bahasa C. *Software* tersebut memiliki kurang lebih 100 *packages* yang berfungsi sebagai algoritma, metode, atau *library*. Dari sudut pandang pemrograman, *software* ini memiliki banyak "fungsi" dan "operasi". Saat ini GAP sudah memiliki lebih dari 800 "fungsi" bawaan untuk mempelajari topik-topik dalam aljabar, sehingga GAP dapat digunakan untuk menyediakan banyak contoh, mulai dari contoh yang sederhana hingga kompleks dalam waktu yang relatif singkat dibandingkan mencari secara manual.

GAP merupakan sistem yang interaktif dan didasarkan pada pengulangan perintah "baca-evaluasi-cetak". Sistem GAP mengambil *input* yang diberikan oleh pengguna, yang diberikan dalam bentuk teks, kemudian mengevaluasi *input* tersebut dan kemudian mencetak hasil evaluasi dari *input* tersebut. Sifat interaktif dari GAP memungkinkan pengguna untuk menulis suatu ekspresi atau perintah dan langsung melihat hasil dari perintah tersebut. Pengguna juga dapat mendefinisikan suatu fungsi dan mengaplikasikannya pada suatu argumen untuk melihat bagaimana fungsi tersebut bekerja.

Kajian teoritis dan literatur yang kami lakukan juga menunjukkan bahwa GAP dapat digunakan dalam pengajaran Struktur Aljabar dimana secara garis besar manfaat *software* GAP antara lain dapat dijelaskan sebagai berikut.

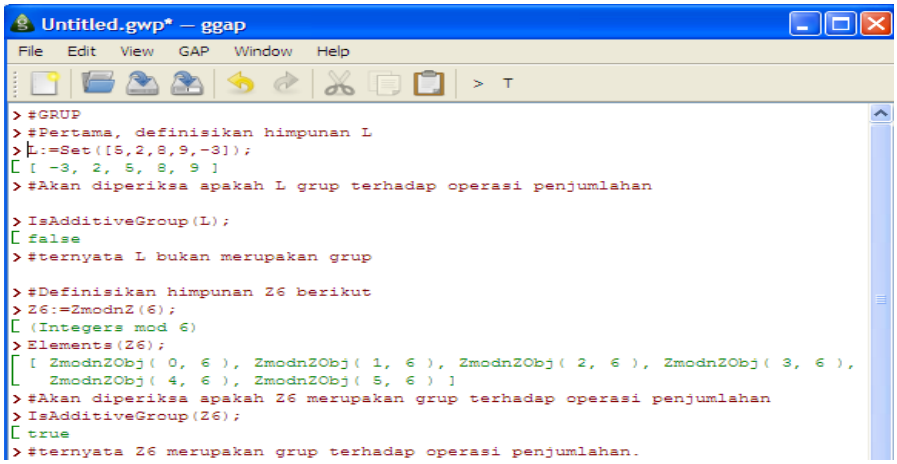
1. Sebagai kalkulator mewah (Rainbolt, 2002)

GAP dapat digunakan sebagai kalkulator yang "mewah" dalam artian bahwa GAP tidak hanya dapat berfungsi sebagai kalkulator biasa pada umumnya, namun GAP juga dapat digunakan sebagai kalkulator untuk menghitung suatu bilangan yang sangat besar. Hal ini dapat dilihat pada Gambar 10.

2. Sarana membuat algoritma sederhana

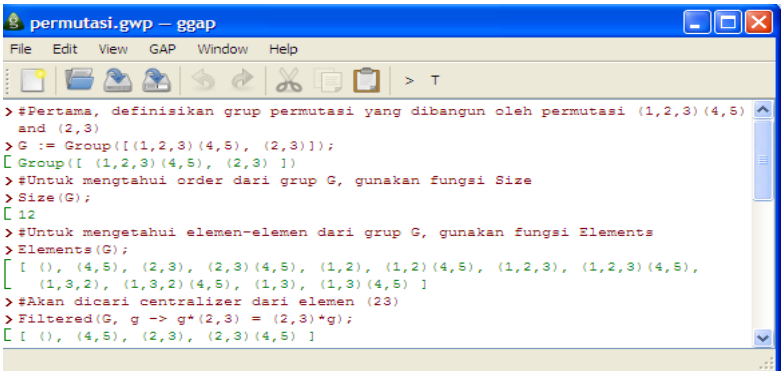
Dengan menggunakan GAP, mahasiswa dapat dilatih untuk membuat algoritma sederhana yang dapat membantu memperkuat pemahaman konsep Struktur Aljabar serta membuat mahasiswa lebih memahami lagi konsep yang baru.

Sebagai contoh, mahasiswa ditugaskan untuk mengkonstruksi suatu himpunan dan kemudian mahasiswa ditugaskan kembali untuk memeriksa apakah himpunan yang mereka konstruksi tersebut merupakan suatu grup terhadap suatu operasi atau bukan. Seperti yang dapat terlihat pada Gambar 1, mereka juga dapat memeriksa apakah suatu himpunan yang telah umum dikenal sebagai $\mathbb{Z}_6$ merupakan suatu grup terhadap operasi penjumlahan.

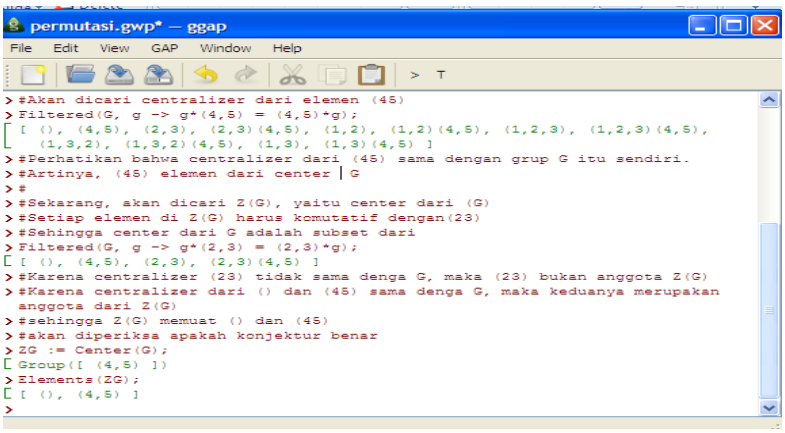


Gambar 1. Konstruksi himpunan dan pemeriksaan strukturnya

Selain hal tersebut di atas, mahasiswa juga dapat membuat suatu grup yang dibangun oleh suatu permutasi. Misalnya mahasiswa ditugaskan untuk membuat suatu grup yang dibangun oleh permutasi (1 2 3)(4 5) dan (2 3), seperti terlihat pada Gambar 2. Selanjutnya, mahasiswa ditugaskan untuk mencari semua elemen dari grup tersebut, baik dengan menggunakan cara manual, maupun menggunakan fungsi bawaan pada GAP. Setelah mengetahui elemen-elemen dari grup tersebut, mereka kemudian dapat mencari ordenya. Dari grup yang dibentuk, dapat dicari *centralizer* dari masing-masing elemen dan *center* dari grup tersebut, seperti yang dapat dilihat pada Gambar 3.

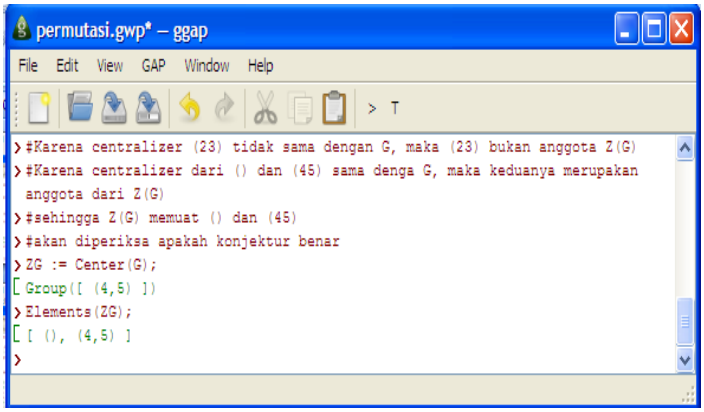


Gambar 2. Konstruksi grup yang dibangun oleh (1 2 3)(4 5) dan (2 3)



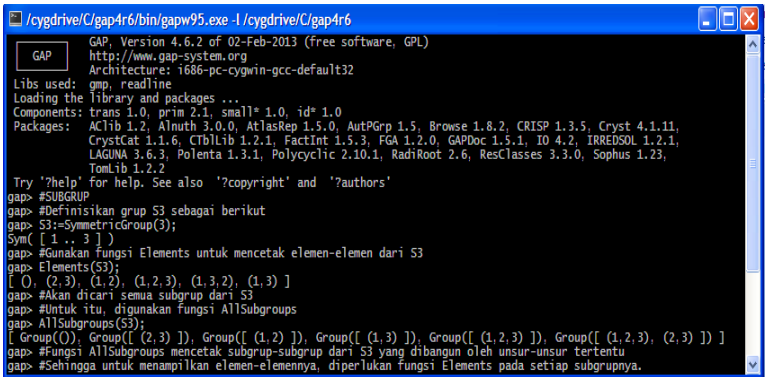
Gambar 3. Mencari *centralizer* dan *center*

Berdasarkan hal-hal yang telah dilakukan di atas, dapat dimunculkan pertanyaan untuk mahasiswa, contohnya apakah suatu elemen dari grup merupakan elemen dari *center*-nya. Di sini mahasiswa dilatih untuk membuat suatu *conjecture* yang kebenarannya dapat diperiksa baik dengan pembuktian secara manual melalui teori-teori yang telah diberikan, ataupun dengan menggunakan GAP (Gambar 4).



Gambar 4. Menampilkan *center* dari suatu grup

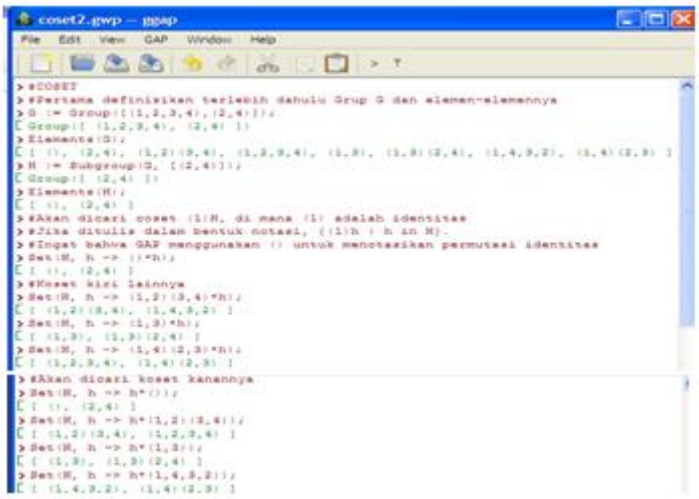
Untuk memahami konsep sub grup, setelah dosen memberikan definisi dan contoh-contoh dari sub grup secara manual, mahasiswa dapat diberi tugas untuk mencari sub grup-sub grup dari grup yang lain dengan menggunakan GAP. Dalam hal ini, misalkan mahasiswa ditugaskan untuk mencari semua sub grup dari grup simetri S_3. Pada Gambar 5 dijelaskan algoritma pencarian semua sub grup dari grup simetri S_3. Proses dimulai dengan mendefinisikan grup S_3. Kemudian, dengan fungsi bawaan GAP yaitu "*AllSubgroups*" (Gallian, 2010) dapat dicari semua sub grup dari grup S_3 tersebut. Untuk lebih meyakinkan bahwa hasil pekerjaan sudah benar, mahasiswa dapat memeriksa kembali sifat tak kosong, himpunan bagian, ketertutupan, dan eksistensi *invers* dari setiap elemen secara bertahap dengan menggunakan GAP. Dengan demikian mahasiswa dapat belajar lebih baik, karena dalam penyelesaian tugas yang menggunakan GAP tersebut mahasiswa akan melalui proses *trial and error* yang membuat mereka menjadi lebih ingat tentang konsep sub grup tersebut.



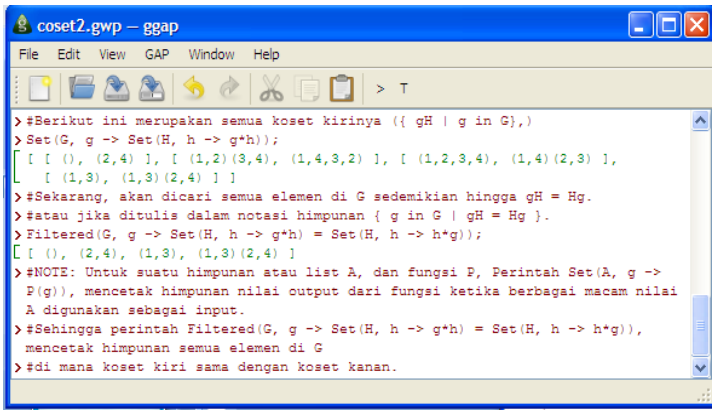
Gambar 5. Mencari semua sub grup dari grup simetri S_3

Dengan adanya suatu grup G dan sebuah sub grup S dari G, akan dibangun grup baru, yaitu grup kuosien G/S dengan syarat S-nya merupakan sub grup normal, yaitu sub grup yang memiliki koset kiri sama dengan koset kanan. Sehingga dalam penggunaan GAP setelah mahasiswa diperkenalkan konsep grup dan sub grup, selanjutnya dapat dilatih membentuk koset kiri maupun koset kanan. Gambar 6 memberikan gambaran langkah-langkah

pengerjaan yang dapat dilakukan oleh mahasiswa. Pertama, mahasiswa ditugaskan untuk membentuk suatu grup dan sub grup yang dibangun oleh suatu permutasi. Setelah itu, mahasiswa ditugaskan untuk mencari semua koset kanan dan koset kiri dengan menggunakan GAP berdasarkan definisi yang telah mereka dapatkan. Mereka akan mengoperasikan semua elemen dari grup dengan sub grupnya. Hasil yang diperoleh menunjukkan terdapat beberapa koset yang sama, sehingga mereka bisa mengidentifikasi koset yang berbeda. Banyak langkah percobaan yang harus mereka lakukan untuk memperoleh koset yang berbeda tersebut, memunculkan pertanyaan bagaimana bila ingin mendapatkan semua koset-koset tersebut hanya dalam satu baris perintah GAP (Gambar 7). Selanjutnya mahasiswa ditugaskan untuk memeriksa apakah koset kiri dan koset kanannya sama. Hal tersebut, selain membuat mereka lebih memahami materi koset yang telah diperoleh, dapat pula dilatih untuk membuat algoritma sederhana.

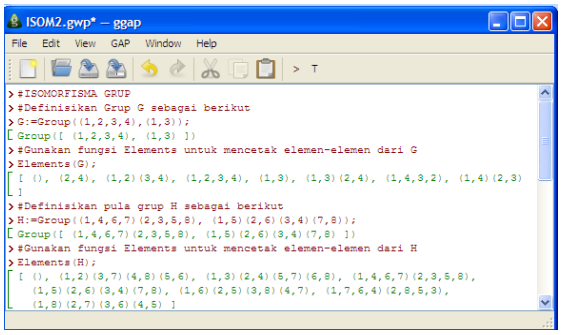


Gambar 6. Mencari koset kiri dan koset kanan



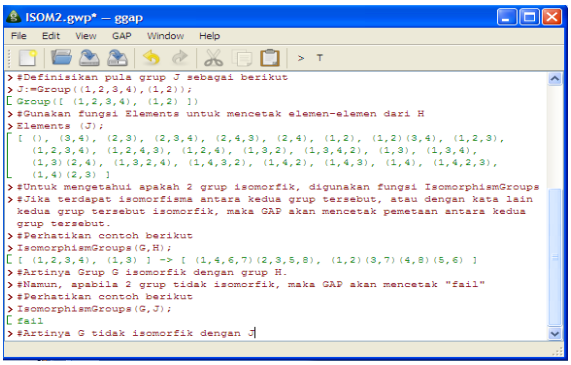
Gambar 7. Memeriksa kesamaan koset kiri dan koset kanan

Materi terakhir dalam mata kuliah Struktur Aljabar 1 adalah teorema isomorfisma. Untuk memahami konsep tersebut dengan menggunakan GAP mahasiswa ditugaskan untuk mengkonstruksi dua macam grup (Gambar 8).



Gambar 8. Mengkonstruksikan dua macam grup

Mahasiswa kemudian ditugaskan untuk mendefinisikan suatu pemetaan antara kedua grup tersebut dan memeriksa secara manual apakah pemetaan itu merupakan suatu isomorfisma. Setelah mahasiswa dapat mengerjakan hal tersebut, dapat ditunjukkan bahwa dengan menggunakan GAP dapat diketahui secara langsung apakah terdapat isomorfisma antara kedua grup tersebut (Gambar 9).



Gambar 9. Isomorfisma Grup

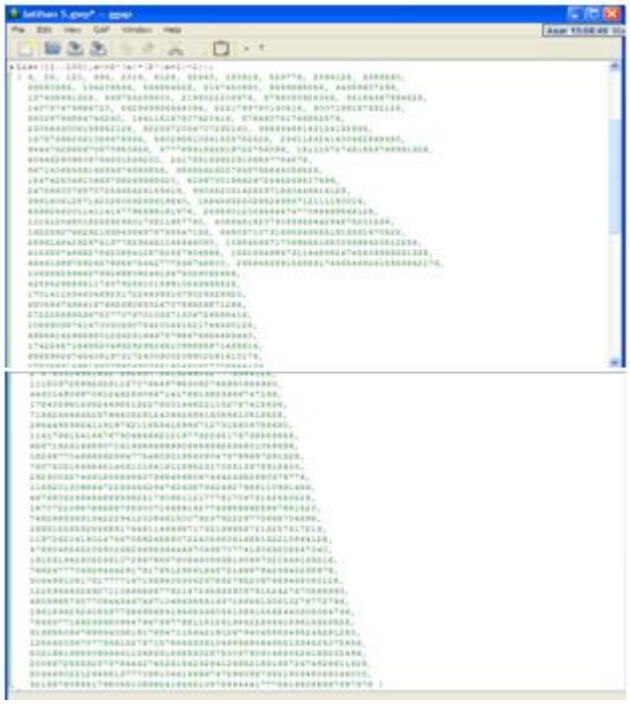
Dari uraian di atas terlihat bahwa penggunaan GAP pada pembelajaran Struktur Aljabar diharapkan dapat memberikan motivasi untuk mempelajari mata kuliah tersebut dengan cara yang tidak membosankan, yang pada akhirnya dapat meningkatkan pemahaman mahasiswa terhadap mata kuliah tersebut.

GAP juga dapat digunakan sebagai alat untuk memproduksi sejumlah data yang besar secara cepat, sehingga mahasiswa dapat melihat pola dari data tersebut dan kemudian membangun suatu *conjecture*. Seperti dalam proses pencarian *perfect number*, yaitu suatu bilangan integer positif dan merupakan hasil penjumlahan dari semua pembagi sejati bilangan itu sendiri (Andrews, 1971), mahasiswa dapat membuat suatu *conjecture* syarat cukup bagi suatu bilangan yang merupakan *perfect number*.

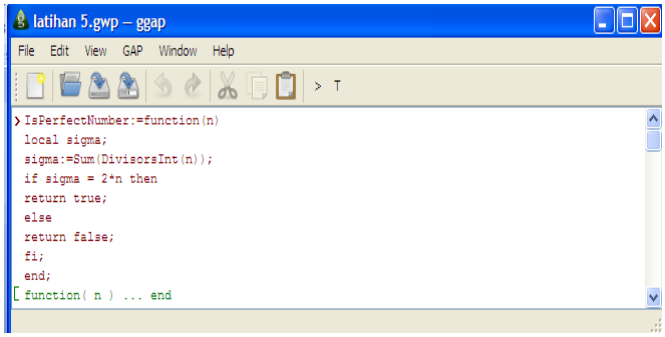
Sebagai contoh, misalnya mahasiswa ditugaskan untuk menghitung $2^n(2^{n+1}-1)$ untuk $n = 1, 2, \ldots, 100$. Mahasiswa kemudian ditugaskan untuk mencari *perfect number* dari hasil perhitungan tersebut. Selanjutnya mahasiswa ditugaskan kembali dengan tugas yang serupa dengan mengubah n menjadi lebih besar, misalnya, $n = 1, 2, \ldots, 1000$. Setelah itu, mahasiswa ditugaskan untuk membuat kesimpulan apakah semua bilangan yang berbentuk $2^n(2^{n+1}-1)$ selalu merupakan *perfect number*? Ataukah ada kondisi tertentu yang membuat hal tersebut tidak berlaku? Kondisi apakah yang dapat membuat $2^n(2^{n+1}-1)$ menjadi *perfect number*? Hasil *conjecture* tersebut dapat dibuat dalam suatu teorema yang dapat dibuktikan kebenarannya.

Dari tugas tersebut, hal yang pertama dilakukan oleh mahasiswa adalah menghitung

$2^n(2^{n+1}-1)$ untuk $n = 1, 2, \ldots, 100$ dengan menggunakan GAP. Untuk mencari *perfect number* dari bilangan-bilangan yang ditampilkan sebagi hasil perhitungan GAP dapat dilakukan secara manual. Namun, karena data yang dihasilkan sangat besar dan banyak, maka pencarian dengan manual tidak efisien. Oleh karena itu, diperlukan untuk membuat suatu fungsi dalam mencari *perfect number* dari data yang sudah dimiliki.

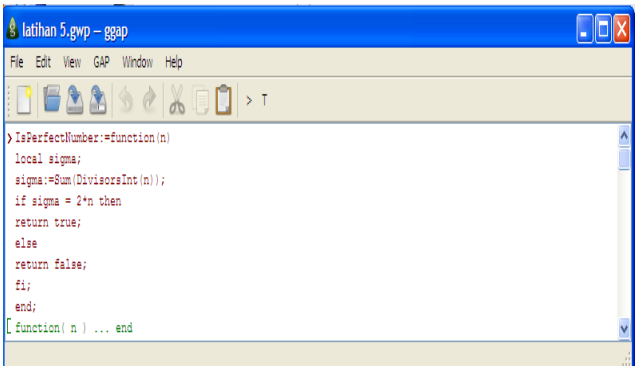


Gambar 10. Hasil perhitungan $2^n(2^{n+1}-1)$ untuk $n = 1, 2, \ldots, 100$



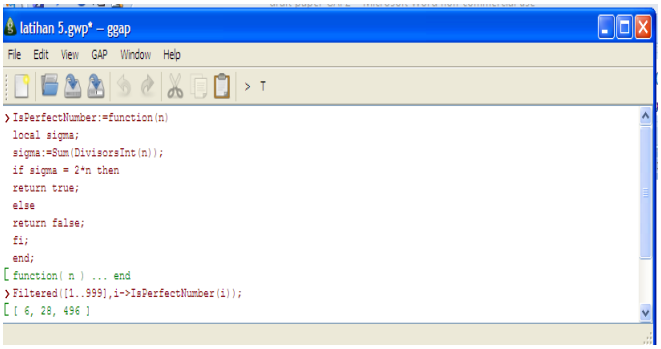
Gambar 11. Fungsi pengujian *Perfect Number* pada GAP

Untuk membuat fungsi tersebut, mahasiswa ditugaskan untuk membuat definisi *perfect number* dalam bentuk fungsi pada GAP. Fungsi yang dibuat adalah fungsi untuk menguji apakah bilangan yang di*input*kan merupakan *perfect number*.



Gambar 12. Fungsi *Perfect Number*

Setelah fungsi *perfect number* dibuat, langkah selanjutnya adalah memilah bilangan-bilangan yang bukan merupakan *perfect number*. Untuk itu, diperlukan operasi "*Filtered*" (Hulpke, 2011). Dengan menggunakan "operasi" tersebut, maka bilangan yang tercetak hanyalah bilangan-bilangan yang merupakan *perfect number*. Dengan penggunaan GAP, selain memahami suatu konsep Aljabar, mahasiswa juga diajak untuk dapat menulis konsep tersebut dalam suatu algoritma.



Gambar 13. Hasil penyisihan *Perfect Number* pada GAP

Dari contoh pencarian *perfect number* di atas, kita dapat melihat bahwa mahasiswa diajak untuk dapat membangun suatu teorema atau *conjecture* atau menemukan contoh yang terkadang juga dapat menjadi suatu pekerjaan yang sulit. Contoh di atas juga memperlihatkan bahwa kita dapat membangun teorema dengan mengasumsikan suatu kondisi hingga dan menggunakannya dalam *GAP*.

KESIMPULAN

Dari penjelasan di atas, dapat disimpulkan bahwa manfaat-manfaat yang dapat diperoleh dari penggunaan GAP antara lain:

1. Sebagai kalkulator "mewah" yang dapat menghitung bilangan yang cukup besar dan dapat dituliskan dalam digit yang lengkap,
2. Dapat membantu mahasiswa untuk memahami suatu konsep Struktur Aljabar,
3. Dapat membantu mahasiswa untuk menyelidiki ataupun membangun suatu *conjecture*, yang pada akhirnya mengarahkan mahasiswa untuk membuat suatu teorema yang dapat dibuktikan kebenarannya,
4. Dengan demikian penggunaan GAP diharapkan dapat memberikan motivasi untuk belajar Struktur Aljabar dengan cara yang menyenangkan yang pada akhirnya dapat meningkatkan penguasaan mahasiswa terhadap mata kuliah tersebut.

DAFTAR PUSTAKA

Andrews, G.E. (1971). *Number Theory*. Philadelphia: W.B. Saunders Company

Direktorat Akademik Ditjen Dikti. (2008). *Buku Panduan Pengembangan Kurikulum Berbasis Kompetensi Pendidikan Tinggi (sebuah alternatif penyusunan kurikulum)*. Jakarta: DIKTI

Gallian, J. A. (2010). *Abstract Algebra with GAP for Contemporary Abstract Algebra 7th edition.* Boston: Brooks-Cole Cengage Learning.

Hulpke, A. (2011). *Abstract Algebra in GAP*. California: The Creative Commons Attribution-Noncommercial-Share Alike 3.0

Rainbolt, J.G. (2002). *Teaching Abstract Algebra with GAP*. Saint Louis.

The GAP Group. (2006). *GAP – Groups, Algorithms, and Programming*, Version 4.4, [Online] [Diakses dari http://www.gap-system.org.]